



Docket No. 40921-250098 (1301-008)  
PATENT

In Re: Throop, Dean

Serial No. 09/752,199

Filed December 29, 2000

Examiner Joshua D. Schneider

Art Unit 2182

Title METHOD AND SYSTEM FOR ENCODING SCSI REQUESTS FOR  
TRANSMISSION USING TCP/IP

**DECLARATION OF DEAN THROOP UNDER 37 C.F.R. 1.131**

I, Dean Throop, the inventor in the above-identified application being duly warned, hereby declare as follows:

I conceived the invention described and claimed in the above-identified application at least as early as August 1, 1999, and that work on reducing the invention to practice commenced on or about October or November of 1999.

The invention was reduced to practice through development and operated successfully at least as early as January of the year 2000.

Evidence of the dates set forth previously and with respect to conception and reduction to practice may be found in the appended completed invention disclosure form used by the assignee of this application. This invention disclosure form was completed by me in the ordinary course of record keeping which is part of my responsibilities as an employee of the assignee of this application.

Further evidence of conception and reduction to practice may be found in the appended paper entitled "Flare Network Interface to Navisphere" circulated internally as a confidential document by the assignee of the invention and above-identified dated October 6, 1999, and describing the invention disclosed in the above-identified application.

The above set forth dates of conception and reduction to practice of the invention are well before the May 31, 2000 filing date (effective date) of U.S. Patent No. 6,836,830 to Yamagami et al., which is being applied to reject the claims in the above-identified application.

I further declare that being warned that willful false statements and the like are punishable by fine or imprisonment, or both (18 U.S.C. 1001), and may jeopardize the validity of the application or any patent issuing thereon, that all statements made of my own knowledge are true and that all statements made on information and belief are believed to be true.

Respectfully Submitted,

  
\_\_\_\_\_  
Dean Throop

Dated: April 26, 2006

Enclosure



\*\*\*\*\*

INSTRUCTIONS

PLEASE COMPLETE THIS DOCUMENT, INSERTING THE REQUESTED INFORMATION AFTER EACH QUESTION. MAIL THE COMPLETED FORM, AND ANY ACCOMPANYING ELECTRONIC DOCUMENTS, VIA EMAIL TO ANGELA MUISE IN THE LAW DEPARTMENT. IF ANY SUPPORTING MATERIALS ARE BEING SUBMITTED IN HARD COPY, PLEASE PRINT A COPY OF THIS FORM, ATTACH TO THE HARD COPY MATERIALS AND MAIL TO ANGELA MUISE, LAW DEPARTMENT, 35 PARKWOOD DRIVE, HOPKINTON.

\*\*\*\*\*

I. TITLE OF INVENTION:

Encoding SCSI Requests for Transmission using TCP/IP

---

II. INFORMATION ABOUT INVENTOR(S):

1. Name: Dean Throop

Home Address: 112 Stratford Dr

Chapel Hill NC 27516

Badge No.: 11221 Mail Stop: rtp Citizenship: U.S.

(If more than one individual contributed to the invention, supply the above information for each additional inventor.)

III BACKGROUND INFORMATION

1. Will a DG/EMC product use this invention?  
YES

If so, what is the internal name of the DG/EMC product incorporating this invention? Flare for Alpine Phase 2

Has the product been publicly announced by DG/EMC?  
I don't think so

If announced, what name/model is the product marketed under?  

If not yet announced, what is the anticipated announce date for the product? May 5, 2000 (scheduled ship date)

2. On what date did the idea of the invention first come to mind? Aug 1999

3. Other than discussions among the inventors named above, to whom was the invention first disclosed: \_\_\_\_\_

The date of that disclosure: \_\_\_\_\_

I discussed this with Stephen Todd (1<sup>st</sup> in Aug 1999)

It has been implemented by Lorenzo Bailey, Bob Frazier, and David Pressley.

4. On what date was the idea of the invention first reduced to writing in whole or in part? (This would include any form of handwritten, typed or electronic record. Send a copy of this material.) \_\_\_\_\_

There's a specification written October 1999

5. Has construction/development of this invention started? \_\_\_\_\_ Started October or November 1999

If so, date started: \_\_\_\_\_

Has the invention been built/developed and operated successfully? \_\_\_\_\_ Its been working since januay of 2000

If so, date of first successful operation: \_\_\_\_\_

6. Has this invention been disclosed outside of DG/EMC? \_\_\_\_\_ I don't think so

If so, were the disclosures under a non-disclosure agreement? \_\_\_\_\_

Give dates, locations, and other relevant information about all disclosures outside of DG/EMC: \_\_\_\_\_

7. List all functional specs, design specs, management presentations and other available materials that describe or discuss the design or operation of this invention.

\_\_\_\_\_ See attached

#### IV. TECHNICAL DISCLOSURE OF INVENTION

1. State in general terms the nature of the invention.

An encoding of SCSI requests on TCP/IP.

2. Describe any old method(s) of performing the function of the invention, indicating the disadvantages and problems with the old methods that are overcome by the invention. (Note: Describe all prior methods of which the inventors have knowledge, not just prior DG/EMC methods. It is sufficient to state the current knowledge and understanding of the inventors. It is not necessary that the inventors conduct research to identify all "old" methods.)

SCSI requests are generally sent over wires created explicitly for just that. Two common implementations use parallel copper wires and fibre channel lines. The wires are connected using special purpose hardware boards (SCSI controllers) plugged into computers.

The operating system running on the computers has software that uses the SCSI controllers for doing industry standard block I/O. Because the operating systems have exclusive use of the SCSI controllers, it is often difficult to write a user program that can send vendor specific SCSI requests using the SCSI controllers.

The vendor specific requests are useful for configuring the target SCSI device and for monitoring it. Monitoring to observe failures and to track performance.

Some operating systems provide a user pass through capability that allows a user program to send vendor specific SCSI requests to a target device. These pass through capabilities are operating system specific and don't always work very well. Some times Operating system SCSI pass through requests will stop other traffic until they complete (thus degrading performance). Some operating systems do not provide any SCSI pass through mechanism at all.

Another alternative for a user program to access a target device is to connect to it using a serial cable. (We do this right now with clarion). The user program encodes the vendor specific SCSI requests and transmits them to the target device over a serial line. This is available over a wide variety of operating system.

Unfortunately this encoding is error prone and rather slow.

By encoding SCSI requests and transmitting them using TCP/IP, we have a fast and widely available way for user applications to send vendor specific SCSI request to a target device.

2. Describe the design and operation of the invention, indicating the specific features believed to be new and all advantages or improvements over the old methods. Use drawings, schematics, timing diagrams, flowcharts, and/or sketches as appropriate for a clear and complete understanding of the invention.

The specification is available on the web at

[http://thiinman.rtp.dg.com/~flare/projects/flare\\_projects.htm](http://thiinman.rtp.dg.com/~flare/projects/flare_projects.htm)  
You will need username=clarion, password=diskus

->alpine network  
->Flare Network Interface to Navisphere-2

4. Are you aware of alternative ways in which the invention could be used or implemented? If so, describe generally (a) other applications for the invention (for example, other types of products that could benefit from the invention) and (b) alternative ways the invention could be constructed or implemented.

There is a draft for a similar implementation.

A URL for this Internet-Draft is:

<http://www.ietf.org/internet-drafts/draft-satran-iscsi-00.txt>

I've written a brief comparison of our approach with the above draft and our implementation and I'll attach that.

Name of Person Preparing This Disclosure: Dean Throop

Date of Disclosure: \_\_\_\_\_



# Flare Network Interface to Navisphere

Dean Throop

October 6, 1999

## 1 Introduction

This proposes a revised interface to Flare to allow the Navisphere Agent to manage Flare over TCP/IP.

The original Alpine Phase II Ethernet Service Port Functional Specification proposed using the existing serial line protocol over a TCP connection. However upon further consideration this approach was found unappealing because of synchronization problems with the current serial line protocol and the number of serial line specific manipulation calls made by the Agent.

A better interface would be to pass SCSI requests over TCP. This would standardize the management interface on the SCSI CDBs already defined and treat TCP/IP as an alternative transport to the Fibre Channel interface.

This means it will be possible to issue arbitrary SCSI requests such as read or write over the network. In general read or write requests will not be issued over the network as Fibre is the optimal interface for that. However there doesn't seem to be any reason to prohibit doing exactly that. It might be a useful to be able to generate read or write requests for testing. It might also be nice to have a back door to access LUN data for test verification or backup. It doesn't make sense to use it as the primary data storage data path but the ability will be there for completeness.

## 2 Overview

The goal of this new interface is to treat each TCP connection from a Navisphere Agent to Flare as a separate SCSI initiator. The Navisphere Agent can send multiple SCSI requests sent over the TCP connection and those requests will be processed according to SCSI rules for reordering and parallelism. Replies can come back in any order.

Completely modeling SCSI over Fibre on TCP would create some inefficiencies so the interface will be somewhat modified. To completely model SCSI over Fibre, the Agent would send a request to Flare and then Flare would request transfer of the data buffer associated with the request. This would mean an extra round trip message for Flare to read a data buffer. Rather than having Flare send requests to retrieve a data buffer from the Agent, the Agent will always send the data buffers that will be read by Flare as part of the request. For data buffers that are written by Flare, Flare will return these to the Agent as soon as processing has completed the buffer. When the Agent sends a request, TCP flow control may prevent the Agent write operation from completing immediately. If the Agent has several requests outstanding at one time, it should always keep accepting data (have a separate thread always doing a read or equivalent) so Flare can return completed requests to free resources to process incoming requests.

Having the Navisphere Agent send the data buffer with requests in which Flare will read the data buffer means the buffer will be transferred unnecessarily if the SCSI request has an error. Errors are expected to be infrequent and the saving of an extra round trip message should more than offset this.

## 3 References

- [1] CLARiiON Disk-Array Licensed Internal Code Specification, Full-Fibre Storage Systems, Drawing No: 009-001854-04
- [2] Storage Centric Setup Command Interface Rev 1.3  
Charlie Hopkins 5/7/99

## 4 Environment

### 4.1 Security

Flare will only service connections from the Navisphere agent if the IP address of the agent matches one of a configured set of addresses that Flare will trust. The Flare administrator must configure the list of addresses as part of Flare management. Each address will have an associated mask to allow the administrator the ability to match an entire subnet with a single entry.

The Flare network Interface only offers a modest level of security. Administrators can telnet to Flare where access is granted to anyone presenting the proper password. The password will be passed over the network in clear text so anyone on the network with a sniffer can obtain the password. If someone obtains the Flare password, they can access Flare and unbind LUNs and zero disks thus destroying data. Sites that are concerned about security should only connect Flare to a physically secure LAN segment trusted for management use and protected from other access (the site should use a firewall or take equivalent precautions to limit network connectivity).

Because the address from which Flare will accept connections must be specified in advance, the hosts running the Navisphere Agent must use fixed IP address; they can not use DHCP or some other dynamic mechanism of obtaining an address. This limitation results from the use of static addresses to grant access. If this proves to be a serious problem, we'll consider adding a more flexible mechanism in the future.

## 4.2 Network Performance

If Flare is connected to a network with a lot of broadcast traffic, that traffic could cause a performance impact on Flare. Flare should only be connected to a network where the level of broadcast traffic is known to be low enough to not adversely impact the performance of hosts on that network.

## 4.3 Encoding

The SCSI requests will be encoded using CTLD as described in the Storage Centric spec[2]. SCSI requests will be encoded with tags. The fields will be structured similarly to fibre requests and will include a Request Id to allow the Agent to match replies with requests. Data buffers transferred from Flare to the Agent will be returned independent of, and may proceed the status of the request that generated the data.

The CTLD encoding must always use the Binary encoding. All CDBs and data buffers will be in network byte order just as they would be constructed for the Fibre interface. Request Ids must be 4 bytes long.

## 5 Messages

### 5.1 SCSI Request

This message sends a SCSI request from the Agent to Flare.

Tag = TAG_NET_SCSI_REQUEST
Sub Tag = TAG_NET_REQUEST_ID
Sub Tag = TAG_NET_SCSI_CMND_PAYLOAD
Sub Tag = TAG_NET_BUFFER

The TAG\_NET\_REQUEST\_ID is uninterpreted by FLARE and will be returned in the reply message.

The TAG\_NET\_SCSI\_CMND\_PAYLOAD contains a FCP CMND Payload as described in the CLARIION Fibre Spec[1].

The TAG\_NET\_BUFFER is a conditional Sub Tag; if Flare will read the data buffer from the Agent for the CDB, this provides that data. If Flare will return a data buffer to the Agent, this Sub Tag should not be present.

### 5.2 SCSI Reply

This message returns the status of a SCSI request from Flare to the Agent.

Tag = TAG_NET_SCSI_REPLY
Sub Tag = TAG_NET_REQUEST_ID
Sub Tag = TAG_NET_SCSI_RSP_PAYLOAD

The TAG\_NET\_REQUEST\_ID returns the Request Id of the request.

The TAG\_NET\_SCSI\_RSP\_PAYLOAD returns a FCP RSP Payload as described in the Clarion Fibre spec[1].

### 5.3 SCSI Buffer

This message returns the data generated by Flare when processing a SCSI request. It may proceed or follow the SCSI status for the request.

Tag = TAG_NET_SCSI_RESULT_BUFFER
Sub Tag = TAG_NET_REQUEST_ID
Sub Tag = TAG_NET_SCSI_BUFFER

The TAG\_NET\_REQUEST\_ID returns the Request Id of the request that generated this reply buffer.

The TAG\_NET\_SCSI\_BUFFER returns the data buffer generated by Flare.

### 5.4 Option Request

This message sends a TCP specific request from the Agent to Flare. At this time there are no options defined. At some time in the future the TCP interface may support network specific behavior such as having Flare generate Alerts for some conditions. All the Sub-tags other than the Request Id tag of this will be ignored in the initial version of this interface.

Tag = TAG_NET_OPTION_REQUEST
Sub Tag = TAG_NET_REQUEST_ID

The TAG\_NET\_REQUEST\_ID specifies a Request Id tag that Flare will return with the reply.

This option request can be sent at any time. In the future if the interface is changed and the old interface no longer supported, the option request maybe required to be the first message to allow Flare and the Agent to agree to use the new interface; an appropriate Tag will be defined with the new interface.

Other tags maybe added in the future as options are defined.

### 5.5 Option Reply

This message returns the result of an Option Request from Flare to the Agent. At this time the response will always be unsupported request. At some time in the future this request may be defined and the reply will return a status of zero to indicate successful request.

Tag = TAG_NET_OPTION_REPLY
Sub Tag = TAG_NET_REQUEST_ID
Sub Tag = TAG_NET_OPTION_STATUS

The TAG\_NET\_REQUEST\_ID returns the Id of the request that generated this reply.

The TAG\_NET\_OPTIONS\_STATUS returns a value of 1 indicating request not supported.

This message will also be sent by Flare to inform the Agent of problems with the connection. If Flare receives a connection and finds it can not service the connection, Flare will send an Option Reply

message with a status indicating why it will not service the connection. Thus if the Agent were to send an Option Request as the first request to Flare, a reply with a failure status will indicate why the connection is being closed.

The following values are defined for the status:

- 0 Request accepted
- 1 Request not supported (or tag in request not supported)
- 2 Connection not accepted, authorization failure
- 3 Connection not accepted, too many connections

## **6 Connection management**

Flare will listen for TCP connections on port 2918. It will accept all connections. If Flare already has more than the implementation maximum (at least 8), it will send a TAG\_NET\_OPTION\_REPLY with a status value of 3 indicating that the maximum number of connections has been exceeded and then close the connection. If Flare will not accept the connection because the agent's IP address is not in the list of allowed hosts, Flare will send a TAG\_NET\_OPTION\_REPLY with a status value of 2.

When reading connections, Flare will implement a timeout. If Flare ever waits more than 100 seconds to finish reading a complete message, it will close the connection. The timeout does not start until the first byte of the message has been received; thus an idle connection can remain indefinitely. Flare will enable TCP keepalives on the connection to cleanup connection from hosts that are no longer accessible.

If Flare ever finds an invalid length on a Request Id, or SCSI CDB, it will assume the Agent and Flare are out of sync due to some problem and it will close the connection.

## **Appendix A: TAG LIST**

TBD

End of Document